

Using the John Oliger Expansion System to Create Instant-Load (EPROM) Software for TS/1000 Computers

Part III.



Tony Gomez

I. Introduction

In part II last month, we showed you how to program two 2764 eproms, one with HOT-Z, the other with the Oliger machine code downloading routine.

DOWNLOAD MC UTILITY

```

C009 CDE702      CALL 02E7
C00C 2100C0      LD HL,C000
C00F 110B00      LD DE,000B
C012 19          ADD HL,DE
C013 5E          LD E,(HL)
C014 23          INC HL
C015 56          LD D,(HL)
C016 3A2140      LD A,(4021)
C019 A7          AND A
C01A EB          EX DE,HL
C01B 2019        JR NZ,C035
C01D 010940      LD BC,VER3N
C020 C5          PUSH BC
C021 ED42        SBC HL,BC
C023 44          LD B,H
C024 4D          LD C,L
C025 EB          EX DE,HL
C026 110C00      LD DE,000C
C029 ED52        SBC HL,DE
C02B D1          POP DE
C02C EDB0        LDIR
C02E E1          POP HL
C02F 217606      LD HL,0676
C032 E5          PUSH HL
C033 C30702      JP 0207
C036 011540      LD BC,4015
C039 ED42        SBC HL,BC
C03B 19          ADD HL,DE
C03C FD3521      DEC (IY+21)
C03F 7E          LD A,(HL)
C040 FEFF        CP FF
C042 20CB        JR NZ,C00F
C044 CF          RST 08H
C045 0D          ERROR E
C046 00          NOP
C047 00          NOP
C048 00          NOP
C049 00          NOP
C04A 00          NOP
C04B 00          NOP
C04C 00          NOP
C04D 00          NOP
C04E 00          NOP
C04F 00          NOP

```

END

Now, we are ready to program our final eprom with "CETCH2". This is my program which demonstrates some of the features of the Oliger Video Upgrade. A

general procedure will be discussed first, followed by the specific example: "CETCH2".

II. General Procedure

Before powering up, turn on the 48-64K area of RAM, and prepare the 2764 eprom programmer with a completely erased 2764 eprom.

The program that you wish to "burn" into the 2764 eprom must be already "SAVED" to tape. If you don't need the variables, CLEAR the program before "SAVEing" to tape to save eprom space.

Now, if you have prepared your HOT-Z eprom, carefully place it in EPROM B location on Video Board B. If you have not, then you will have to reload HOT-Z on cassette form.

Assuming that "HOT-Z" is resident in the 8-16K space, and that your program is SAVED on tape, call HOT-Z by whatever means is available to you. If you have the eprom version, you get it by calling RAND USR 8192. If you have the cassette, then load the HOT-Z cassette, and then go to the disassembly mode following successful loading.

HOT-Z is now in its READ or disassembly mode. Next, enter the next number C009. This is decimal address 49161. This is the start location for the temporary storage of your BASIC program. Note that when you have entered C009, you will see data "NOPs", indicating that there is no data there yet. Enter "Y" which gets you into HEX-EDIT MODE.

Next, enter "P", which sets the end-of-program (EOP) cursor. Enter the nex number FFFF (65535d), because this is the last possible byte of 64K ram available. Prepare your cassette player with the program to be loaded. Enter "J", which sets up the HOT-Z load routine. Do not enter the program name or double quotes. Hit ENTER and play the tape. HOT-Z will then proceed to load your program into the space between C009 and FFFFh (49161- 65535d).

If you are using the Oliger Video Upgrade, the border will alternate colors during loading. Eventually, the screen will indicate that loading has ceased. (no flashing borders). Hit "BREAK" key to return to HOT-Z. Note that now there is data in the 48-64K space. This is your newly loaded program.

Now, enter the nex number C014, and observe the disassembled data present at C014 and C015. Write down this nex data. Now, form a new hex number by taking the data contents of C015 and C014-1, where the latter is one hex value less than the data found at C014. This new number formed is the theoretical end of the program relative to the 16-32K RAM space, but since we did not load anything in there with HOT-Z, we must find the number relative to the 48-64K RAM space, where it truly is. Add 8000h to this new number. Enter this new hex number with HOT-Z. Using the "K" switch, find the decimal equivalent. Note: if the answer is greater than 57343, then you must do the programming in two steps. Oliger explains this in his notes.

All that remains is to quit HOT-Z, and enter a simple BASIC program which will actually program the 2864 eprom with the contents of high memory (what HOT-Z has just loaded in).

If all this has been just a bit too much to grasp, the following example will help clear things up. My basic program "CETCH2", which creates color patterns on the screen (using) Oliger's Video Upgrade), will be loaded into high memory for eventual burning onto an erased 2764 eprom. See listing 1 below. Enter the program in listing 1 by keyboard, CLEAR, then SAVE it to tape.

III. Example: Programming a 2764 eprom with "CETCH2"

Call HOT-Z by whatever means available.
Enter C009 in HOT-Z disassembly mode.
Enter "Y" to set the cursor at C009.
Enter "P" to set the end-of-program cursor.
Enter FFFF to set the end of program value (65535d).
Enter "J" to set up the HOT-Z tape loading routine.
Play "CETCH2" program tape. Don't enter title or double quotes. Just press "ENTER".
Observe that the program is being loaded in (flashing borders).
After loading is complete, (no border activity), hit "BREAK" which will return you to disassembly mode.
*Enter C014 and observe the following data:
C014 = B7h C015 =
4Ah Note that C014-1 = B6h.
Form the new number C015 + (C014-1) =
4AB6h.
Add 8000h to this number to obtain CAB6h.
Enter CAB6h with HOT-Z and hit "K" for data display to obtain the decimal equivalent of 51894.

This address, 51894, is the address of the last byte of program as loaded by hot-Z. Escape HOT-Z by hitting "Q", and enter the following BASIC program:

*To verify accuracy of loaded program, perform a

CHECKSUM as we did in part II last month, except change line 20: 20 FOR N = 49161 to 51894. The answer must be CHECKSUM = 106038.

```
10 SLOW
20 LET X = 8192
30 FOR N = 49161 to 51894 (the last address
   found above)
40 POKE X, PEEK N
50 PAUSE 4
60 LET X = X + 1
70 NEXT N
80 PRINT "PROGRAMMING COMPLETE"
90 PRINT X; " = NEXT AVAILABLE BYTE"
```

Before running the program, make sure that the VPP programmer is set to +21 volts. Hit ENTER, and the eprom programmer will perform its task. The programming time for "CETCH2" is about 4½ minutes.

From the program, one can see that every byte of data from 49161 to the end of the program at 51894 is being programmed into the 2764, beginning at 8192. The PAUSE 4 is vital because there is a minimum delay between programming consecutive addresses in the 2764. The 2764 is burned everytime the POKE command is given.

When finished, the display will indicate the message and the next available byte for programming. That means the next program you may wish to put on the current EPROM must begin immediately at this address. If you do have a new program that you wish to put immediately following "CETCH2", then in line 20 we must change the EPROM starting burn address as follows:

```
LET X = 10926
```

Since everything up to that address is already programmed with CETCH2. If the new program is longer than the theoretical bytes remaining on the current eprom, then the programming must be accomplished in two steps. John Oliger goes into this in his notes.

IV. Demonstration Program Use

At last you have finished programming all your 2764 eproms. Now it's time to use them. A few introductory words are necessary, however.

The MC Download eprom should be placed into EPROM B socket on Video Board B. Turn EPROM B switch ON. The EPROM with "CETCH2" must be placed into the EPROM A socket on the 2764 reader board, and its switch ON. The 48-64K RAM memory switch must be turned OFF. The 8-16K RAM memory switch must likewise be turned OFF. All these switches are necessary so that there will be no conflicts between RAM and APROM firmware.

Now you must identify the program on the 2764 read board to the computer. Do this by entering:

```
POKE 16417,0
```

What you are doing is identifying the first program on the eprom as "0". In general, POKE 16417,X where X is the number of the program desired to be downloaded. The first program number is 0, the second (if there is one) is 1, etc. Programs must be stored on eprom with no separating bytes.

If the desired program, other than program 0, does not exist in the cartridge, then the computer will return with report "E", program requested is not on cartridge board.

So, following your POKE statement, and after checking all switches for proper set-up,

RAND USR 8192

Which will call the machine code download utility resident in eprom B on video board B, and in an instant "CETCH2" will be downloaded into RAM from 4009h on.

Enter a number between + and -5. "1" is a good starting place. The "etch-a-sketch" pattern will begin. At any time you can change the paper and ink to one of 32 possible combinations by hitting "C". If you wish to see the number which created the particular paper/ink combination hit "S". The program will momentarily stop, and print the number at the bottom of the screen.

Keep in mind that many more paper/ink combinations are possible, but I programmed in those which I felt gave the best color contrast. See John Oliger's notes on how to create other combinations.

V. Conclusion

I hope this practical demonstration has interested you into a new, exciting way to enter programs on the Oliger expansion system. If you liked the use of HOT-Z and machine code in general, I'd like to recommend a new publication which goes into machine code programming on a very practical level: "A PRACTICAL GUIDE TO MACHINE LANGUAGE PROGRAMMING ON THE TIMEX/SINCLAIR 1500/1000/ZX-81 COMPUTERS" by David B. Wood, for \$14.95 + \$1.00 shipping/handling. It is available from SIRIUSWARE, 6 Turning Mill Road, Lexington, Mass. 02173. The book was recently reviewed (Oct'84) in SYNTAX. (Current ad is in CTM.)

I mentioned previously those of you interested in verifying if your 2764 eproms were completely erased could obtain a cassette program to do this. The program is entitled "PROMERASE", and comes with complete instructions. It even gives additional details on how to produce an eprom version for instant call up. It's available for \$10.00 — postage, shipping, and handling included. Send your check or money order (preferred) to:

Tony Gomez
2013 Los Feliz Apt 3
Thousand Oaks, Ca 91362

Next time we get together, I plan to talk about the new expansion system John Oliger has recently introduced for the 2068 computer. Until then, have

fun and enjoyment with your expanded systems.

CTM

BEST PRICES ON MOBILE ANTENNAS and ACCESSORIES

EXAMPLE

5/8 WAVE 2 METER
MAGNET MOUNT ANTENNA
TUNED TO 146MHz
COMPLETE with 15 FEET COAX
AND CONNECTOR
\$18.95

QUARTER WAVE 2 METER MAGNET MOUNT
COMPLETE \$12.95

H.F. MOBILE MONO BAND HELICAL
\$12.95

Add \$3.00 per order for shipping

Send SASE for catalog to:

GEORGE SHIRA WD4BUM
RT #7 101-I
ANDERSON, SC 29624
(803) 226-6990

Now With Windowing! \$49.95 Basic Compiler

MTBASIC

Features:

Multitasking	Windowing
Handles interrupts	Interactive
Fast native code	Compiles quickly
Floating point	No runtime fee

MTBASIC is a true native code compiler. It runs Bytes's Sept. '81 seive in 26 seconds; interpreters take over 1400 seconds! Because MTBASIC is multitasking, it can run up to 10 Basic routines at the same time, while displaying ten separate windows. Pop-up/down menus are a snap to implement.

MTBASIC combines the best of interpreters and compilers. To the programmer, MTBASIC appears to be an extremely fast interpreter. MTBASIC compiles a typical 100 line Basic program in 1 second, yet it generates blindingly fast code. No more waiting for long compiles.

AVAILABLE for CP/M (Z-80) and PC-DOS systems.

ORDERING: Specify format when ordering. We accept Visa, MC, checks and COD. Send \$49.95 plus \$3.50 shipping and handling (\$10 overseas) to:



P.O. Box 2412 Columbia, MD 21045-1412
301/792-8096